Jaws ERS

Gary Davidian x4510, 27AJ, DAVIDIAN1
Bob Herold x5118, 27AJ, HEROLD1
Revision 0.1   April 17, 1989

Revision History

Rev 0.0	11/4/88	ggd	first release
Rev 0.0 D1	1/18/89	ggd	updated for Blue Book
Rev 0.1	4/17/89	rwh	reworked for hype, current thinking

What is Jaws

Jaws is a software project to improve the quality of our ROM sources. It consists of the following efforts:

- a rework of ROM sources to make them "Universal", enabling a single ROM image to run on many hardware platforms. Two universal ROMs are envisioned: Jaws32 to run all 68020/030 based machines, and Jaws16 to run all 68000 based machines.

  The first release of Jaws32 would support Aurora, F19 and the Mac II family (II, IIx, IIcx and SE/30) without Big Bang code in ROM. The next release of Jaws would be a new Jaws 32 that adds support for Pearson and Eclipse (040), and a Jaws16 that supports our 68000 machines and possibly Elsie (a hot topic these days).

- a rework of ROM sources to make them easier to port to new hardware platforms

- a rewrite of select routines in the ROM to improve performance

Problems that Jaws Solves

Software maintenance is hard

As of April 1989, we are shipping three ROM versions: ROM 75 in the Mac Plus, ROM 76 in the Mac SE, and ROM 78 in the Mac II, IIx, IIcx, and SE/30 (actually the IIx, IIcx and SE/30 have ROM 78 rev 1.3). Whenever we patch in a new software feature across all machines, it usually must be implemented separately for each ROM version. The effort involved is multiplied by the number of ROM versions.

Each ROM version has its own unique features. Bugs are discovered that occur in a particular version, or in some of the versions, or in all of them. The complexity of finding and fixing ROM bugs increases combinatorially with each new ROM version.

Apple must produce new versions of the ROM to support new hardware platforms and to move more of its software base into the legal protection of the ROM. However, new ROM versions will only make maintenance harder. Reducing the number of new ROM versions will reduce the future additional overhead imposed on adding new features to and fixing bugs in the ROM.

ROM ports take too long

Over the years Macintosh software development has typically been hardware project driven. New features were added to the ROM to be released with a particular piece of hardware. Examples include HFS for the MacPlus, ADB for the MacSE and Color Quickdraw for the Mac II.

Code for new features was usually added to the ROM enclosed by conditional assembly directives. However, the directives selected what *machine* the ROM was being built for, instead of selecting what *software feature* was actually being added. This makes including those new features in a ROM port for a new machine difficult. Massive editing is needed to add the new machine's identifier to all conditional assembly statements around desired features.

Changing the source to use feature based conditionals makes porting much easier. Instead of editing the source, a build script for the new machine can be set up enabling the conditionals for the particular software features desired.

Performance is not what it can be

Much of the Macintosh source is designed to run on our entire hardware line. It must therefore run on an 68000, a 68020 and a 68030. Often this means a performance sacrifice on 68020 and 68030 machines, as these processors have new instructions, addressing modes and optimizations that cannot be used.

Additionally, the original Macintosh ROM was optimized for space, sometimes at the expense of speed. It had to fit in 64k!

Finally, countless engineers have been hacking away at the ROM sources for five years now. While their intentions may have been noble, their legacy is often less-than-optimal code. When projects are near completion, it is often safer to fix a bug in a non-optimal way rather than rewrite a large section of code and risk introducing more bugs.

Jaws Goals

Given all these problems to solve, the goals of the Jaws project are to:

1) Reduce the number of new ROM versions by enabling the same ROM image to support many machines.

2) Reduce the porting effort to new hardware platforms by cleaning up the sources.

3) Improve ROM software performance

Benefits of Jaws

If we reach our goals for the project, we will realize the following benefits:

1) Easier maintenance. With the number of new ROM versions reduced to a minimum, software engineers will spend less time retrofitting a new feature to existing machines. Bug fixing will be easier as well.

2) Faster ROM ports. This benefits result from two of the Jaws efforts. First, the ROM will be able to identify hardware features and configure itself for that hardware. No porting effort is needed for old hardware features in new machines. Second, software features will be included with feature based conditionals. They can be included in a new ROM by creating one new MPW build script, instead of editing hundreds of source files.

3) Testing effort can be simpler. Since the same ROM image will run on multiple hardware platforms, functional testing of the hardware independent code will be transferable across the platforms. While there are no guarantees that the same code on two different machines will produce identical results, we can still be much smarter about where we expend our testing resources to gain the most benefit.

4) The ROM sources will be easier to maintain. Given that we will be looking at entire files of code, rather than tweaking a few lines here and there, we will clean up the result of five years of hacking. More readable source will result.

5) Our machines will be faster. Optimizing for the 68020/030/040 in select ROM routines can dramatically improve performance. Given that we are producing a 68000 version and a 68020/030 (and later 040) version, these optimizations are now possible.

6) We can reduce developer seeding costs by sending them new ROMs for their existing machines instead of new hardware. For most developers this will be adequate. Certain developers will need the actual hardware, e.g. F19's for Localtalk products such as TOPS, or Aurora's for monitor vendors.

7) Manufacturing costs may be reduced since a single set of ROM parts can be stocked for many different machines.

What Makes a ROM Jaws-Like

Jaws will allow a single ROM image to be used on several different hardware configurations. When the machine is first booted, the ROM code will determine which hardware features are available on the machine it is running on, and configure itself to use those features. If different code is required to handle dissimilar implementations of a hardware feature, both copies of the code will be included and the appropriate code will be selected at runtime. A good example of this is the Aurora/MacII implementation of the Sony Driver where the main processor manipulates the SWIM chip directly, and the F19 implementation where an IOP (Input Output Processor) manipulates the SWIM chip.

Configuration will take place in many ways, depending on the situation. For some hardware dependent code, jump vectors may be initialized to point to the correct code. In other areas, the code will check a flag on entry to see which hardware it is running on, and take the appropriate path.

Writing Jaws-Friendly Code

It is highly desirable that most software be independent of the hardware it is running on. Exceptions include device drivers and low level operating system code such as the interrupt handler and the start code. Macros will be available for these exceptional cases to test for specific hardware configurations. We will attempt to hide as much as possible the gritty details of how to determine what hardware is being used.

A separate document, 'Writing Code for a Jaws ROM", details how to make your code universal. This document is currently vaporware. If you really need to know now, see one of the authors.

What the Jaws Project Isn't.

While there are a number of new system features that are being introduced with the Jaws ROM (32 bit clean memory manager, on board video drivers, IOP based drivers, SCSI DMA, remote booting, etc.), the implementation of those features is outside the scope of the Jaws project, although the code to configure those features (if they are hardware dependent) is within the Jaws project.

Famous ROMs in Macintosh History

The concept of a universal ROM is not new to the Macintosh. As described below, all of the ROMs released to date have had some sort of universal features.

The very first Macintosh 64K ROM for the original 128K Macintosh (known as ROM 69), was universal with respect to system RAM size, and was also used in the Macintosh 512K.

The Macintosh Plus and Macintosh 512Ke 128K ROM (known as ROM 75), had several universal features. It was available as a field upgrade ROM, along with an 800K floppy drive, for the Macintosh 512K, to transform it into the 512Ke. It supported the following features if they were present.

- Memory size ranging from 512K to 4 megabytes.
- If the 53C80 SCSI controller chip was present, the SCSI manager would support it.
- It supported two different versions of the Real Time Clock / Parameter RAM chip. The original one had 20 bytes of parameter ram, while the newer one had 256 bytes.
- It contained limited support for the Motorola 68010 and 68020 microprocessors, even though Apple never produced a machine using this ROM with those processors.
- The Serial Driver supported two different types of serial port connectors, with different signals. The Mac Plus serial ports supported Data Terminal Ready (DTR), while the 512Ke did not have this signal available.
- The Sony Driver supported 3 type of disk drives, the original single sided 400K drive, the newer double sided 400K/800K drive, and the slow non-SCSI Hard Disk 20.

The Macintosh SE 256K ROM (known as ROM 76), is not very universal. Unlike the Mac Plus ROM, it cannot be used earlier Macintosh hardware. The universal features are as follows.

- Memory size ranging from 1 to 4 megabytes.
- It contained limited support for the Motorola 68010 and 68020 microprocessors, even though Apple never produced a machine using this ROM with those processors.
- The Sony Driver supported 3 type of disk drives, the original single sided 400K drive, the newer double sided 400K/800K drive, and the slow non-SCSI Hard Disk 20.
- A later revision of this ROM has been produced which contains a new Sony Driver that will support the SWIM chip and the FDHD disk drive if they are present.

The Macintosh II 256K ROM (known as ROM 78), was the first ROM for the open architecture Macintosh, it required a 68020 processor, and contained many new features, and was not compatible with any earlier machine. The universal features are as follows.

- Memory size ranging from 1 to 8 megabytes.
- Supported either the HMMU memory mapping unit, or the Motorola 68851 PMMU paged memory management unit.

The Macintosh IIx ROM (also known as ROM 78, rev 1.3), is a fairly universal modification of the original Macintosh II ROM. It can be used in the original Macintosh II, the Macintosh IIx, the Macintosh SE/30, Macintosh IIcx, and is supplied with the SuperDrive upgrade for the Macintosh II. The universal features are as follows.

- Memory size ranging from 1 to 128 megabytes.
- Supported either the 68020 or the 68030 microprocessor.
- Supported the HMMU, the 68851, or the built in 68030 memory management unit.
- Supported either the 68881 or the 68882 floating point coprocessors.
- A new Sony Driver that supports either the IWM or the SWIM disk interface chips, and the FDHD disk drive.
- Support for detecting the Mac II, Mac IIx, Mac SE/30, and Mac IIcx logic boards.

The Jaws 32 ROM - First Release

Initially we will focus on the 32 bit Macintosh family. The Jaws 32 ROM assumes that it is running on at least a 68020 processor, and can be optimized to use the additional instructions and addressing modes that are not available on the 68000. This implementation will be targeted for F19, and may possibly be included in Aurora if the impact is determined to be small.

Below are a list of the kinds of features that the Jaws 32 ROM will support, and the envisioned list of possible implementations to choose from.

This ROM will support all prior machines that were supported by the Mac IIx ROM. This ROM could be used for a field upgrade to 32 bit clean memory manager support for the earlier machines.

- Memory size ranging from 1 to 128 megabytes.
- Supported either the 68020 or the 68030 microprocessor.
- Clock Rates of 16, 20, 25 or 33 Mhz.
- Supported the HMMU, the 68851, or the built in 68030 memory management unit.
- Supported either the 68881 or the 68882 floating point coprocessors.
- Support 680x0 IWM/SWIM, or SWIM IOP based Sony drivers.
- Support 680x0 SCC, or SCC IOP based AppleTalk/Serial drivers
- Support for both Appletalk 1.0 and Appletalk 2.0.
- Implement ADB using either the ADB Transceiver processor, or SWIM IOP based ADB driver.
- Support either the Mac II VIA2, the RBV, or the OSS chips for NuBus and other external interrupts.
- Support RBV based onboard video, or Mac SE/30 onboard video, or no onboard video.
- Take advantage of the SCSI DMA chip if present.
- If an external cache exists, use it. FourSquare and F19 caches are supported.
- Support for detecting the Mac II, Mac IIx, Mac SE/30, Mac IIcx, Aurora, FourSquare, F19, and Spin logic boards.

The Jaws 16 ROM

Since there is only one 68000 based CPU in development (Esprit), and it is very far along, it would be too disruptive to make it a Jaws ROM. We can still begin development of a universal 16 bit ROM that would support all of the 68000 based 16 bit machines. This would be more important if a new low cost Macintosh, or a ROM upgrade for existing machines is being considered.

Much of the effort involved in creating the Jaws32 ROM is beneficial to the Jaws 16 effort. The source cleanup and the separation of hardware dependent code all are beneficial.

It is not anticipated that a Jaws16 effort will begin in earnest until the Jaws32 ROM is solid.